

## **ABSTRACT OF THE DISCLOSURE**

Methods, apparatus and computer program products are disclosed for a method of invoking a native method in a Java virtual machine ("JVM"). A  
5 special-purpose fast interface, executing in conjunction with an interpreter loop, for native methods reduces C stack recursion in the JVM. The interface performs as an extension to the interpreter loop component in the JVM in that a native method, invoked via the special-purpose interface, is able to modify the interpreter loop state if necessary. This is done without adding new  
10 bytecode instructions. A method of executing a native method in a Java virtual machine is described. The JVM first determines whether a native method is to be handled by a special native interface or one of multiple other native interfaces. If it is determined that the method is to be handled by the special native interface, the method is invoked and passed arguments enabling  
15 it to access the state of the JVM. The method is then executed. The state of the JVM is adjusted based on the execution of the method such that transition between the interpreter loop and the native method via the special native interface is minimized.

5  
10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100